

Trajectory Generation for Vehicles Moving with Constraints on a Complex Terrain

Ken-Jui Tsao, Li-Sheng Wang, Po-Ting Kuo
Institute of Applied Mechanics
National Taiwan University
Taipei, Taiwan, ROC

Fan-Ren Chang
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, ROC

Abstract – In this paper¹, the methodology of generating an optimal trajectory on a complex terrain for a specific vehicle is proposed. The possible paths are constrained by the limitations on the terrain and the capability of the vehicle. To deal with these constraints, the notions of forbidden point, forbidden direction, and forbidden path are introduced. After certain constants are specified, the method of dynamic programming is then invoked to find the optimal solution. If the target is beyond the maximal range of the vehicle, appropriate service stations are selected by using the auction algorithm. To speed up the computation process, the ideas of bi-spiral scheme and instant update are employed. With all the techniques at hand, numerical results show that the proposed method can generate the desired trajectory efficiently.

1. Introduction

For a vehicle moving on a complex terrain, its capability restricts the possible paths that it can follow. A jeep can climb a steeper slope than a passenger car, while a mobile robot may be able to climb even steeper slope. Different wear on the tire leads to different limit on the angle of turn. How to accommodate these constraints on the generation of optimal trajectory on a complex terrain is the main theme of this paper.

The constraints on the path may come from either the topology of the terrain or the vehicle itself. On the terrain, there may be some hazardous region or congested area that the vehicle cannot enter. There may be some service stations that the vehicle may need to pass. On the part of the vehicle, sharp turns may not be able to trace and it is preferred to move on a flat path rather than a slope. As a result, the classical algorithm of finding the shortest path needs to be modified. For the shortest-path problem, various techniques have been used to find the optimal solution. These include the Label correcting method [1], Dijkstra's method [3], method of dynamic programming [6][7][8][9], and the auction algorithm [2]. If the dynamics of the vehicle is taken into consideration, various scheme such as the Hamilton-Jacobi's method can be adopted, as reviewed in [10]. However, if the complicated dynamical behavior is considered through certain constraints, for the corresponding path planning problems in which other characteristics, in addition to the length of the path, must be considered, the method of dynamic programming provides the most flexible platform based on the principle of optimality. The idea of dynamic programming has been used in [5] to

find the optimal path of a vehicle on a terrain by including the consideration of forbidden region and the slope. In this paper, similar idea is adopted to find the optimal solution by considering more realistic limitations and the capability of the vehicle.

In particular, the notions of forbidden directions and forbidden paths are introduced in this paper. The limitations of rate of climb, minimal angle of turn, and maximal distance of travel are incorporated into the algorithm. Depending on the slope of the path, the scaled distance is used to perform the optimization process, so that flatter paths are preferred. If the length of the optimal path exceeds a certain bound, the vehicle needs to pass through some service stations. For this problem, we first transform it to a classical salesman problem and then use the auction algorithm to select the suitable service stations. Moreover, to enhance the computational speed, the notion of bi-spiral scheme is introduced and the technique of instant update is applied. From the numerical results shown in this paper, the methodology proposed here can indeed generate an optimal trajectory that meets all the above-mentioned requirements.

The rest of this paper is organized as follows. In Section 2, the underlying problem of path planning on a complex terrain is described in more detail. Section 3 presents the setup for realizing those constraints and the process of incorporating them into the method of dynamic programming. The algorithm for the selection of service stations and the bi-spiral scheme are also discussed. Numerical results are shown in Section 4, which shows the effectiveness of the proposed method. Finally, some concluding remarks are given in Section 5.

2. Problem Description

Consider a complex terrain, such as in Figure 1, consisting of geometric points with each position being specified by three components (x, y, z) , with the corresponding axes being pointing toward the east, the north, and upward, respectively. The horizontal plane on which the terrain resides may be divided into small rectangles with a matrix of grid points (x_i, y_j) , $i = 1, \dots, l$; $j = 1, \dots, n$, which are equally spaced with distance a , cf. Fig. 1. The point on the terrain corresponding to node (i, j) is represented by $P(i, j)$ and the height at that point is given by $z(i, j)$. There may be some regions on the terrain that the vehicle is prohibited to pass, such as a

¹ Corresponding Author: Li-Sheng Wang, email: wangli@gauss.iam.ntu.edu.tw

hazardous area, a lake or a building, which is identified as the set Ω . Moreover, it may be necessary for the vehicle to stop somewhere during the travel for service stations, such as gas stations. The locations of the service stations on the terrain are denoted by nodes g_1, \dots, g_N .

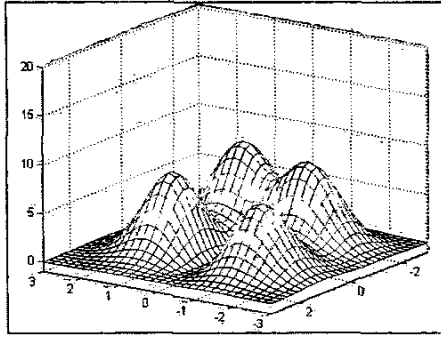


Fig. 1 complex terrain

The problem is to seek efficiently the optimal trajectory from a starting node $s = (i_s, j_s)$ to a target node $t = (i_t, j_t)$ without entering Ω such that the designated vehicle is able to follow. Different types of vehicles, such as a passenger car, a jeep, or a truck, have different capabilities, such that their motions are subject to various constraints. First, the initial direction of the path must be compatible with the heading of the vehicle at the node s . There may be some other nodes for which the direction of passing is limited. Secondly, the vehicle may not be able to move on a steep slope. This renders the limitation on the rate of climb of the trajectory. Thirdly, the vehicle may not be able to make sharp turns, and hence the radius of curvature of the path is constrained. Moreover, the maximum distance of travel may require the vehicle to stop at some service station if the destination is too far to reach with one gas tank. In addition, flat paths are preferred than the paths with steeper slope to save the energy. Other constraints may be imposed on the trajectory of motion. Those listed above shall be accommodated in the generation of the optimal path as discussed in the next section.

3. The algorithm

To solve the problem described above, we adopt the idea of dynamic programming as discussed in [5]. First, we define the forbidden function F as

$$F(i, j) = \begin{cases} 0, & \text{if } (i, j) \in \Omega, \\ 1, & \text{otherwise,} \end{cases} \quad (1)$$

in which Ω is defined in the previous section. In order to describe the move from one node to its neighbors, the virtual move m is defined to be in the set of

$$M^+ = \{1, 2, 3, 4, 6, 7, 8, 9\}$$

such that $m=1$ represents the move from (i, j) to $(i-1, j-1)$, cf. Fig. 2, in which the meanings of the other values of m are also given. Moreover, we use $m=5$ to denote the degenerate virtual move which refers to the move from one node to itself.

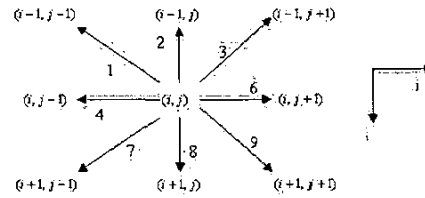


Fig. 2 Virtual Moves

For the virtual move m , the maps φ, ϕ are defined such that $(\varphi(i, m), \phi(j, m))$ is the destined node from (i, j) via m . The distance and the slope of the virtual move m from (i, j) are then computed from, respectively

$$\begin{aligned} dist(i, j, m) &= [a^2(\varphi(i, m) - i)^2 + a^2(\phi(j, m) - j)^2 \\ &\quad + (z(\varphi(i, m), \phi(j, m)) - z(i, j))^2]^{1/2} \\ \alpha(i, j, m) &= \tan^{-1} \left(\frac{z(\varphi(i, m), \phi(j, m)) - z(i, j)}{a\sqrt{(\varphi(i, m) - i)^2 + (\phi(j, m) - j)^2}} \right) \end{aligned}$$

Let the limitation on the rate of climb of the trajectory be represented by the maximal upward slope $\bar{\alpha}$ and maximal downward slope $\underline{\alpha}$, which are determined by the vehicle. To realize the forbidden region, the limitation on the rate of climb, and the preference of the flat paths, the scaled distance function for the virtual move m from (i, j) is defined as

$$D(i, j, m) = \begin{cases} -1, & \text{if } \begin{cases} \varphi(i, m) = -1 \text{ or } > l \\ \text{or } \phi(j, m) = -1 \text{ or } > n \\ \text{or } F(i, j) = 0 \\ \text{or } F(\varphi(i, m), \phi(j, m)) = 0 \\ \text{or } \alpha(i, j, m) > \bar{\alpha} \\ \text{or } \alpha(i, j, m) < \underline{\alpha} \end{cases} \\ (w_0 |\alpha(i, j, m)| + 1) \cdot \text{dist}(i, j, m), & \text{otherwise} \end{cases} \quad (2)$$

in which the scale factor w_0 is specified according to the preference on the slope of the path. It is seen that if the slope of the virtual move m is too large, the distance of the move is set to -1 , which indicates that the corresponding virtual move is prohibited. Moreover, by adjusting the scale factor, the preference of the flatter paths can be realized.

As discussed in the previous section, the directions of passing through some nodes are limited. The set of such nodes, denoted by $b \in B$, contains at least the starting node s . Since the direction of entering b and that of leaving b may have different constraints, we expand the set of virtual moves to include their reversals, denoted by

$$M^- = \{-1, -2, -3, -4, -6, -7, -8, -9\},$$

cf. Fig. 2. The maps φ and ϕ are defined accordingly for $m < 0$. The forbidden directions of passing through the node b are then given by a set $A(i_b, j_b) \subset M^+ \cup M^-$, in which the positive value refers to leaving and negative value refers to entering. If the direction is not allowed, the corresponding virtual move must be prohibited, and the scaled distance function is updated according to the following rule

$$\begin{aligned} & \text{for } m \in A(i_b, j_b), \\ & \text{if } m > 0 \text{ then } D(i_b, j_b, m) = -1 \\ & \text{if } m < 0 \text{ then} \\ & \quad D(\varphi(i_b, m), \phi(j_b, m), |m|) = -1 \end{aligned} \quad (3)$$

Note that for entering forbidden directions, the corresponding virtual moves from $(\varphi(i_b, m), \phi(j_b, m))$ is set to be prohibited.

Next, we characterize the constraint of minimal angle of turn for a designated vehicle. The turn is assumed to be composed of two consecutive line segments between nodes. Consider two consecutive virtual moves m, m' such that

$$(i, j) \xrightarrow{m} (\hat{i}, \hat{j}) \xrightarrow{m'} (\tilde{i}, \tilde{j}).$$

The angle of turn is then given by

$$\begin{aligned} \beta(i, j, m, m') \\ = \angle \overline{P(i, j)P(\hat{i}, \hat{j})}, \overline{P(\hat{i}, \hat{j})P(\tilde{i}, \tilde{j})} \end{aligned}$$

i.e. the angle intersected by the segments $\overline{P(i, j)P(\hat{i}, \hat{j})}$ and $\overline{P(\hat{i}, \hat{j})P(\tilde{i}, \tilde{j})}$. The constraint of turning may be then realized by requiring

$$\beta(i, j, m, m') > \beta_0$$

where β_0 is some constant determined by the type of the vehicle. The path is forbidden if the previous condition is violated.

With the above setup, we now ready to invoke the method of dynamic programming by a recursive process. Since the direction of move is essential in our consideration, we define the total cost starting from (i, j) in the direction of m within k moves by $C(k, i, j, m)$. At the step $k = 0$, the cost function is initialized to be 0 at the node t , and -1 at all other nodes for all virtual moves. As the algorithm proceeds, the value of the cost function is updated according to the following procedure. At step k , at the node (i, j) in the direction m such that $D(i, j, m) \neq -1$, we first construct the set of admissible second moves as

$$\begin{aligned} S(k, i, j, m) \\ = \{m' \in M^+ : \\ D(\varphi(i, m), \phi(j, m), m') \neq -1, \\ C(k, \varphi(i, m), \phi(j, m), m') \neq -1, \\ \beta(i, j, m, m') > \beta_0\}. \end{aligned} \quad (4)$$

The cost function at step $k + 1$ is then found as follows

$$\begin{aligned} & \text{if } S(k, i, j, m) \text{ is empty} \\ & \quad \text{set } C(k+1, i, j, m) = -1 \\ & \text{else} \\ & \quad C(k+1, i, j, m) \\ & \quad = \min_{m' \in S(k, i, j, m)} (D(i, j, m) + C(k, \varphi(i, m), \phi(j, m), m')) \\ & \quad \gamma(k+1, i, j, m) \\ & \quad = \arg \min_{m' \in S(k, i, j, m)} (D(i, j, m) + C(k, \varphi(i, m), \phi(j, m), m')) \\ & \quad \text{if } C(k, i, j, m) \leq C(k+1, i, j, m) \\ & \quad \quad \text{then } C(k+1, i, j, m) = C(k, i, j, m), \\ & \quad \quad \gamma(k+1, i, j, m) = 5 \end{aligned} \quad (5)$$

It is noted that in the second part of the previous procedure, if the above minimization process yields a cost for step $k + 1$ is greater than or equal to that for step k , further virtual move is not appropriate, and hence degenerate move occurs.

The above iteration proceeds until all the virtual moves become degenerate, which is equivalent to the condition,

$$C(\bar{k}+1, i, j, m) = C(\bar{k}, i, j, m), \quad \forall i, j, m.$$

At the final stage, say step \bar{k} , the minimal cost of the optimal path starting from any node (i, j) to the target t in any direction m is obtained. We may then start with s and utilize the map γ to trace back the optimal path as follow:

$$\begin{aligned} m^* &= \arg \min_{m \in M} \{C(\bar{k}, i, j, m)\} \\ i^* &= i_s \\ j^* &= j_s \\ \text{for } q &= 0 \text{ to } \bar{k} - 1 & (6) \\ &\text{if } \gamma(\bar{k} - q, i^*, j^*, m^*) \neq 5 \\ &\text{then} \\ &\quad i^* \leftarrow \phi(i^*, m^*) \\ &\quad j^* \leftarrow \phi(j^*, m^*) \\ &\quad m^* \leftarrow \gamma(\bar{k} - q, i^*, j^*, m^*) \end{aligned}$$

The established path shall then be the optimal one satisfying the constraints discussed in Section 2 except the maximal distance of travel.

To meet the requirement of maximal range R_{\max} for a specific vehicle, we first compute the length of the optimal path found in the above process, say R_{opt} . If $R_{opt} > R_{\max}$, then we need to find suitable intermediate service stations. The strategy is to seek the optimal path for each pair of nodes in the set $\{s, g_1, \dots, g_N, t\}$. Deleting those paths with length greater than R_{\max} , the problem becomes the classical salesman problem as the one shown in Fig. 3. Many shortest-distance methods can be then used to obtain the optimal solution. Here we adopt the auction algorithm as discussed in [2] for its flexibility in dealing with the changes of distance functions.

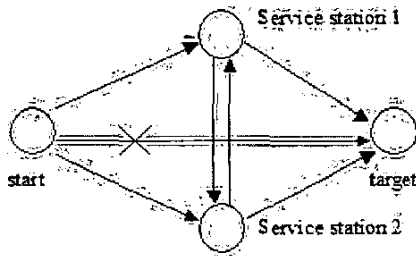


Fig. 3 Salesman Problem and Auction Algorithm

The above process may be very time-consuming if the terrain is large. In order to speed up the computation process, we adopt the bi-spiral scheme in the update process of the cost function. The sequence of nodes visited in each step k to apply the process (5) starts with a neighboring node of the

target t , then proceeds spirally outward. Once all the nodes are visited, the update process reverses its direction to spirally inward return to the target. This sequence can significantly reduce the total number of steps, especially when the desired path is curved back and forth. Moreover, in the minimization process of (5), we could use the updated $C(k+1, \phi(i, m), \phi(j, m), m^*)$, instead of $C(k, \phi(i, m), \phi(j, m), m^*)$ if the node $(\phi(i, m), \phi(j, m))$ has been visited at step $k+1$. This idea of instant update is similar to that of Gauss-Siedel method [4] in solving a set of algebraic equations, and can enhance the efficiency of the algorithm prominently.

4. Numerical Result

For the example of complex terrain shown in Fig. 1, we are now ready to apply the algorithm developed in Section 3 to find the optimal trajectory satisfying the constraints. First, we divide the horizontal plane into 31×31 grid nodes with grid length $a = 5$ (meter), and consider the case that $R_{\max} = \infty$, i.e. the vehicle can move to any point on the terrain without service. Let the capability of the vehicle and the preference of slope impose the constraint on the possible path through the following constants :

$$\bar{\alpha} = 0.7, \underline{\alpha} = -0.8, \beta_0 = 120^\circ, w_0 = 4.0.$$

Incorporating these constants in the algorithm, the optimal trajectory from the node $s = (5, 21)$ to the target $t = (26, 18)$ is then obtained as shown in Fig. 4. The trajectory winds around the hills to reach the target.

If the constraint of maximal range is set as $R_{\max} = 250$ (meter), the vehicle cannot reach the target without service. The strategy outlined in Section 4 using the auction algorithm is then followed to find suitable intermediate service stations. The result is shown in Fig. 5, in which the service stations are marked by big circles. For the same set of the starting node and the target node, another trajectory is chosen to pass by a service station. From these examples, it is seen that the methodology proposed in this paper can indeed give rise to an optimal trajectory satisfying all the constraints described before.

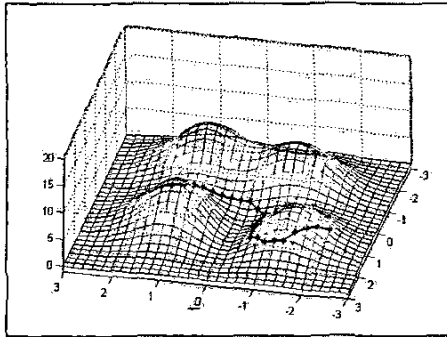


Fig. 4. Optimal Trajectory with Infinite Maximum Range

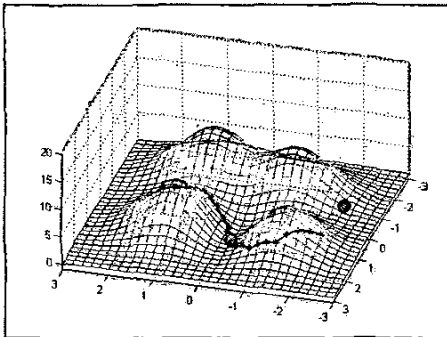


Fig. 5. Optimal Trajectory with Limited Maximum Range

5. Conclusion

This paper presents a methodology to generate an optimal path on a complex 3-D terrain that a specific vehicle is safe and capable to follow. Numerical results show that the method is effective and efficient. The algorithm can be used for a variety of vehicles, such as a passenger car, a truck, or a mobile robot, by choosing the appropriate constants in the algorithm. In addition to those constraints considered here, there may be many other limitations or preference to be included. The effect of the roughness of the terrain on the selection of path is an example. The incorporation of other constraints in the algorithm and the desire of reducing the computational load pave the way of future development of the methodology.

6. References

- [1] Bertsekas, D.P., *Dynamic Programming and Optimal Control*, Athena Scientific, 1995.
- [2] Bertsekas, D.P., "An Auction Algorithm for Shortest Paths," *SIAM J. on Optimization*, vol. 1, pp.425-447, 1991.
- [3] Dijkstra, E.W., "A Note on Two Problems in Connection

- with Graphs," *Numer. Math.*, vol. 1, pp. 269-271, 1959.
- [4] Gerald, C.F., Wheatley, P.O., *Applied Numerical Analysis*, Addison-Wesley Publishing Company, 1994.
- [5] Kwok, K.S., Driessen, B.J., "Path Planning for Complex Terrain Navigation Via Dynamic Programming," *Proc. of American Control Conference, S.D., Cal.*, pp. 2941-2944, 1999.
- [6] Lam, K.P., Tong, C.W., "Connectionist Network for Dynamic Programming Problems", *IEE Proc.-Comput. Digit. Tech.*, vol. 144, No. 3, pp. 163-167, 1997.
- [7] Larson, R.E., Casti, J.L., *Principles of Dynamic Programming - Part II Advanced Theory and applications*, Marcel Dekker Inc., 1982.
- [8] Lewis, F.L., Syrmos, V.L., *Optimal Control*, John Wiley & Sons, 1995.
- [9] Ravindran, A., Phillips, D.T., Solberg, J.J., *Operations Research - Principles and Practice*, John Wiley & Sons, 1987.
- [10] Tsitsiklis, J.N., "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. on Automatic Control*, vol. 40, No. 9, 1995, pp.1528-1538.